مقدمه

قبل از اینکه به سراغ کدنویسی برویم مقدمه ای در مورد برنامه نویسی بانکهای اطلاعاتی را بررسی کنیم.

برنامه نویسی بانکهای اطلاعاتی از پرکاربردترین و پر رونق ترین بخشهای برنامه نویسی به حساب می آید. برای اتصال به بانکهای اطلاعاتی در زبانهای برنامه نویسی ابزارهای مختلفی وجود دارد، به عنوان مثال در زبانهای برنامه نویسی مثل VB و ++VC و ... از ابزاری به نام ADO (و یا قبل از آن DAO) استفاده می شد. به تدریج با ورود .NET Framework مایکروسافت ابزار جدیدی به نام ADO.NET ساخت و به همراه .NET عرضه کرد. بنابراین برای اتصال به بانکهای اطلاعاتی و استفاده از داده های موجود در آنها در زبانهای تحت .NET باید از ADO.NET استفاده کنیم.

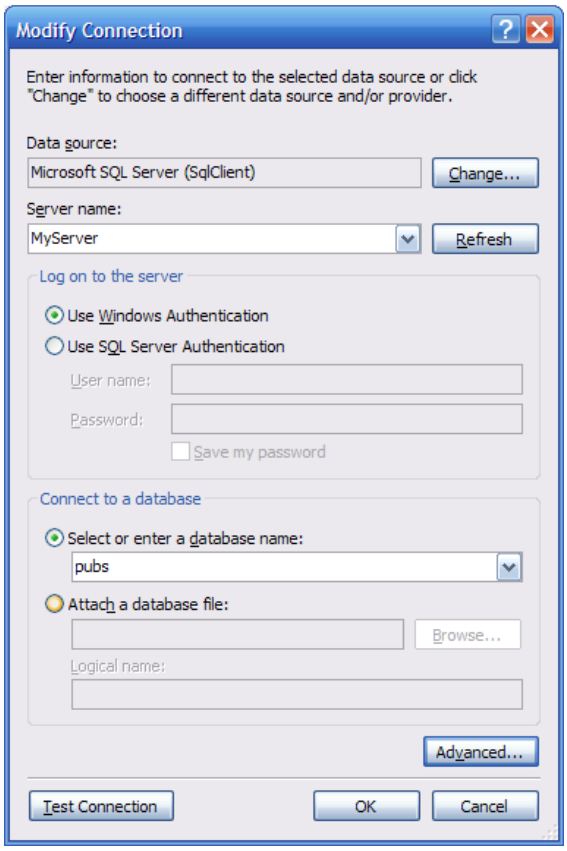
چهار عمل اصلی در کار با بانکهای اطلاعاتی معمولا مورد توجه است که در این سلسله مقالات هم بررسی خواهند شد:

1- خواندن اطلاعات از بانک اطلاعاتی
2- نوشتن اطلاعات در بانک اطلاعاتی
3- تغییر اطلاعات موجود در بانک اطلاعاتی
4- حذف اطلاعات موجود در بانک اطلاعاتی

کلاسهای مورد نیاز برای کار با بانکهای اطلاعاتی با استفاده از ADO.NET در فضای نام System.Data قرار دارد.

اولین چیزی که در برنامه نویسی بانکهای اطلاعاتی باید به آن توجه کنید، نحوه ارتباط با بانک اطلاعاتی مورد نظر می باشد. مشخصات این ارتباط در ConnectionString قرار دارد. در این مقاله فرض بر این است که از بانک اطلاعاتی SQL Server استفاده می کنید. برای مشخص کردن ConnectionString به روش زیر عمل می کنیم:

1- در Server Explorer روی گزینه Data Connection کلیک راست می کنید. (در صورتی که پنجره Server Explorer باز نباشد می توانید از منوی View گزینه Server Explorer را انتخاب کنید).
2- در پنجره Add Connection در قسمت Data Source نوع بانک اطلاعاتی را تعیین می کنیم که در این مثال Microsoft SQL Server می باشد. در قسمت Server Name نام Server مورد نظر را وارد می کنیم. بخش Logon to the Server نوع تعیین هویت کاربر را مشخص می کند. بهتر است از نوع Windows Authentication استفاده نمایید. در بخش Connect to a Database نام بانک اطلاعاتی که در Server تعیین شده وجود دارد را وارد نمایید که در این مثال از بانک اطلاعاتی pubs استفاده می کنیم. این بانک اطلاعاتی بطور پیش فرض با SQL Server 2000 نصب می شود. برای اطمینان از صحت تنظیمات انجام شده روی کلید Test Connection کلیک نمایید.



3- حالا اگر در Server Explorer روی گزینه مربوط به Connection جدیدی که ایجاد کرده ایم کلیک کنید خصوصیات آن در پنجره Properties نمایش داده می شود که از جمله آنها ConnectionString می باشد.

در استفاده از بانکهای اطلاعاتی با استفاده از ADO.NET سناریوهای مختلفی وجود دارد. یکی از این روشها روش Connected یا متصل شده می باشد که در این روش در تمام مدت استفاده از Database ارتباط با آن حفظ می شود. این روش مزایا و معایبی دارد که باعث می شود در بعضی از جاها بتوان از آن استفاده کرد و در جاهایی دیگر استفاده از آن بازدهی برنامه را پایین می آورد. روش دیگر روش Disconnected است که در این روش پس از خواندن اطلاعات از Database ارتباط آن قطع شده و در زمان لازم دوباره ارتباط برقرار می شود.

در اینجا برنامه نویسی در حالت Connected را بررسی می کنیم. در این روش به سه شی از کلاسهای SqlConnection و SqlDataReader و SqlCommand نیاز داریم. به پیشوند sql که در ابتدای آنها آمده است توجه کنید. مایکروسافت سه Namespace برای Provider های مختلف (ابزار فراهم کننده امکان دسترسی به بانک اطلاعاتی یا در واقع رابط بین برنامه شما و بانک اطلاعاتی که شما با استفاده از ADO.NET دستورات مورد نظرتان را به Provider تحویل می دهید) به همراه .NET Framework عرضه کرده است. یک Namespace عمومی به نام Oledb که برای اغلب Database ها از آن استفاده می شود. کلاسهای این Namespace با پیشوند Oledb آغاز می شود. Namespace دوم کلاسهایی هستند که برای کار با Database های SQL Server بهینه شده اند. نام کلاسها در این Namespace با Sql شروع می شوند. Namespace سوم برای کار با بانک اطلاعاتی Oracle تهیه شده است. در اینجا از کلاسهای مربوط به SQL استفاده می کنیم.

هدف از انجام این برنامه اتصال به بانک اطلاعاتی Pubs (که بطور پیش فرض در SQL Server 2000 وجود دارد) و خواندن اطلاعات موجود در جدول Jobs و نمایش مقادیر ذخیره شده در فیلد Job_Desc در یک ListBox می باشد.

برای این کار ابتدا یک پروژه از نوع Windows Application بسازید و سپس یک ListBox و یک Button در آن قرار دهید. نام ListBox را به lstJobs و نام Button را به btnGetJobs تغییر دهید. خاصیت Text مربوط به Button را هم برابر "Get Jobs" قرار می دهیم. اندازه فرم و کنترلهای روی آن را مطابق دلخواه خود تنظیم کنید.

در رویداد btnGetJobs_Click کد زیر را بنویسید:

```
string strConnectionString =
      @"Data Source=MyServer;Initial Catalog=pubs;Integrated
Security=True";
string strCommandText="SELECT * FROM Jobs ORDER BY Job_Desc";

System.Data.SqlClient.SqlConnection oConnection = null;
System.Data.SqlClient.SqlCommand oCommand = null;
System.Data.SqlClient.SqlDataReader oDataReader = null;

oConnection = new System.Data.SqlClient.SqlConnection(strConnectionString);
oCommand = new System.Data.SqlClient.SqlCommand(strCommandText);
oCommand.CommandType = System.Data.CommandType.Text;
oCommand.Connection = oConnection;

try
{
      if (oConnection.State != ConnectionState.Open)
            oConnection.Open();
      oDataReader = oCommand.ExecuteReader();
      lstJobs.Items.Clear();
      int iIndex = oDataReader.GetOrdinal("Job_Desc");
      while (oDataReader.Read())
      {
```

```
            lstJobs.Items.Add(oDataReader[iIndex].ToString());
        }
        if (!(oDataReader.IsClosed))
            oDataReader.Close();
}
catch (System.Exception ex)
{
        System.Windows.Forms.MessageBox.Show(ex.Message);
}
```

ابتدا دو متغیر رشته ای تعریف کرده ایم که یکی حاوی ConnectionString است و یکی دیگر حاوی دستورات SQL مورد نیاز برای استخراج اطلاعات از Database. سپس سه متغیر تعریف کرده ایم که به ترتیب عبارتند از:

oConnection: یک شی از جنس System.Data.SqlClient.SqlConnection می باشد که برای اتصال به Database از این شی استفاده می کنیم.

oCommand: برای اجرای دستورات SQL روی بانک اطلاعاتی از این شی استفاده می کنیم. نوع آن System.Data.SqlClient.SqlCommand می باشد.

oDataReader: این شی برای خواندن و حرکت در رکوردها استفاده می شود. نوع آن System.Data.SqlClient.SqlDataReader می باشد. در مورد آن در بخشهای بعدی مقاله حتما صحبت خواهیم کرد. وقتی که در برنامه های مرتبط با Database صحبت از DataReader شود به معنای استفاده از سناریوی Connected می باشد.

پس از تعریف متغیرها شی oConnection و oCommand را می سازیم. در زمان ساخت شی oConnection رشته حاوی ConnectionString را به آن ارسال می کنیم. در زمان ساخت oCommand رشته حاوی دستور SQL را به آن ارسال می کنیم. از آنجایی که در این مثال قصد استفاده از یک دستور SQL داریم و آن را از طریق برنامه می خواهیم به Database ارسال کنیم، نوع محتویات رشته ارسال شده به oCommand را از نوع System.Data.CommandType.Text تعیین می کنیم. دو حالت دیگر آن TableDirect (برای خواندن محتویات یک جدول بطور مستقیم می باشد و رشته ارسالی حاوی نام جدول است) و StoredProcedure (برای اجرای یک روال ذخیره شده در Database از آن استفاده می شود و رشته ارسالی حاوی نام روال ذخیره شده است) می باشند.

در خط بعد Connectionی که باید oCommand از آن برای ارتباط با Database استفاده کند را تعیین می کنیم.

با بررسی وضعیت Connection در صورتی که باز نشده باشد با استفاده از متد oConnection.Open() اتصال به Database را برقرار می کنیم.

DataReader شی ای است که برای استفاده از آن باید از شی Command کمک گرفت. بدین منظور یکی از متدهای شی Command که مربوط به اخذ اطلاعات از Database می باشد را فراخوانی می کنیم. در این مثال متد ExecuteReader از شی oCommand فراخوانی شده است که خروجی آن از جنس DataReader می باشد.

برای اطمینان از خالی بودن ListBox متد lstJobs.Items.Clear() را فراخوانی می کنیم. سپس با استفاده از متد oDataReader.GetOrdinal("Job_Desc") اندیس محل قرار گرفتن فیلد Job_Desc را بدست می آوریم و آن را در

متغیر iIndex ذخیره می کنیم. متد oDataReader.Read() یک رکورد از DataReader را می خواند بنابراین با استفاده از یک حلقه while تا زمانی که مقدار برگشتی این متد برابر false نشده است مقادیر درون DataReader را خوانده و آنها را در ListBox اضافه می کنیم. برای خواندن مقدار فیلد Job_Desc از رکورد جاری به روش زیر عمل می کنیم:

```
oDataReader[iIndex].ToString()
```

در انتها وضعیت DataReader را بررسی می کنیم و در صورتی که بسته نشده باشد با استفاده از متد Close() آن را می بندیم.

در صورتی که کد خود را درست نوشته اید با خیال راحت آن را اجرا کرده و از برنامه خود لذت ببرید.

این مقاله اولین بخش از سلسله مقالاتی است که برای آشنایی با نحوه استفاده از بانکهای اطلاعاتی در C# نوشته شده است و در بخشهای بعد با جزییات بیشتری در این زمینه آشنا خواهیم شد.